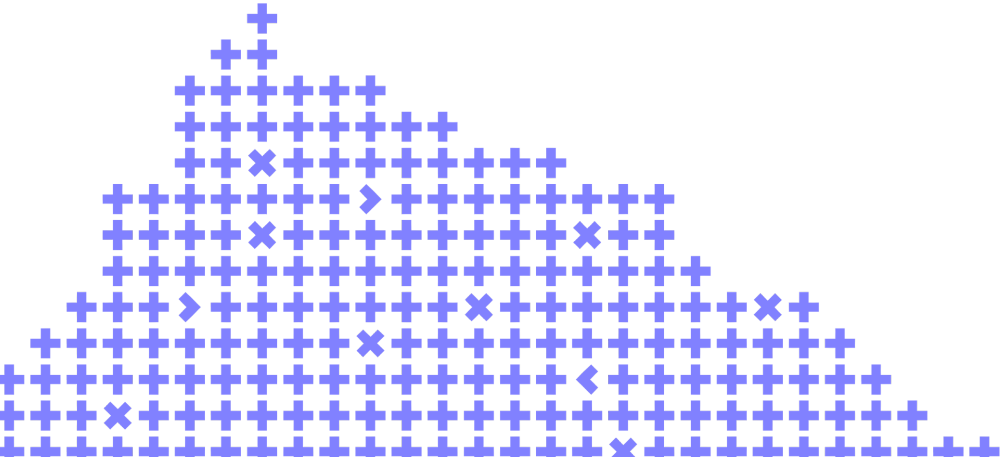


DOCUMENTATION AS A WAY NOT TO FAIL IN MICROSERVICES

PAVEL LAKOSNIKOV



Co-organizer

Yandex

DOCUMENTATION

As a way not to fail
in micro services

Pavel Lakosnikov

Avito Team Lead



1500

microservices

Written on different
languages

2000+ 5

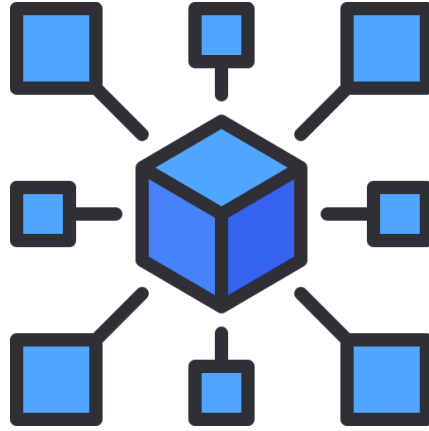
engineers

With different skills,
grade, experience.
Working on different
business domains

5

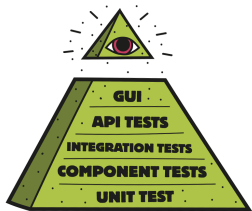
K8S clusters

In different data
centers, in different
parts of the country



With the growth in the number of services, the role of documentation grows

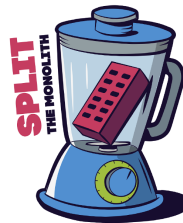
WHY IT'S IMPORTANT



Easy integration
between services



Team
onboarding



Much safer to make
diffs in other teams
services



In fact there are much more bonuses...
But we will talk about later...

WHAT IS — GOOD DOCUMENTATION



ON INTERVIEW

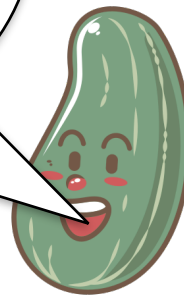


ON INTERVIEW



You

Do you have documentation on
your projects?
Is it good?
Can I use it?
Will it help me to onboard?

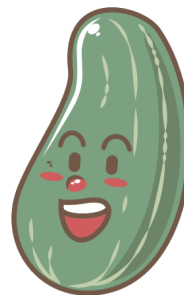


Candidate

ON INTERVIEW



You



Candidate

A composite image of Earth from space, showing continents and oceans. Overlaid on the Earth is a complex network of glowing white lines that arc across the globe, connecting various points. These points are represented by red and blue circular nodes of varying sizes. The background sky transitions from a deep blue on the left to a bright orange and yellow on the right, suggesting a sunset or sunrise. The overall theme is global connectivity and digital networks.

I ASK THAT QUESTION IN THE NET:

I have asked in linkedin, telegram, VK.

Perhaps you see some *strange* guy,
asking about documentation in
different chats and topics....

That was me







We need some *analysis*...and a bit of *group by*

DESCRIPTION

You want to integrate with the service

- ▶ Name, Description
- ▶ Feature Toggles
- ▶ Owner, team, duty
- ▶ QualityGates



QUALITY GATES



It will help us understand the cost of integration and predict quality of our design



Examples:

- ▶ Test Coverage
- ▶ Required e2e tests
- ▶ Linter status

INTEGRATION

- ▶ Service API
- ▶ Integration guide
- ▶ Service NFR + SLO\SLA



Non-functional Requirements

Describe every handler in [nfr.toml](#) artefact

- Max\Min RPM
- Latency
- Expected Error Rate

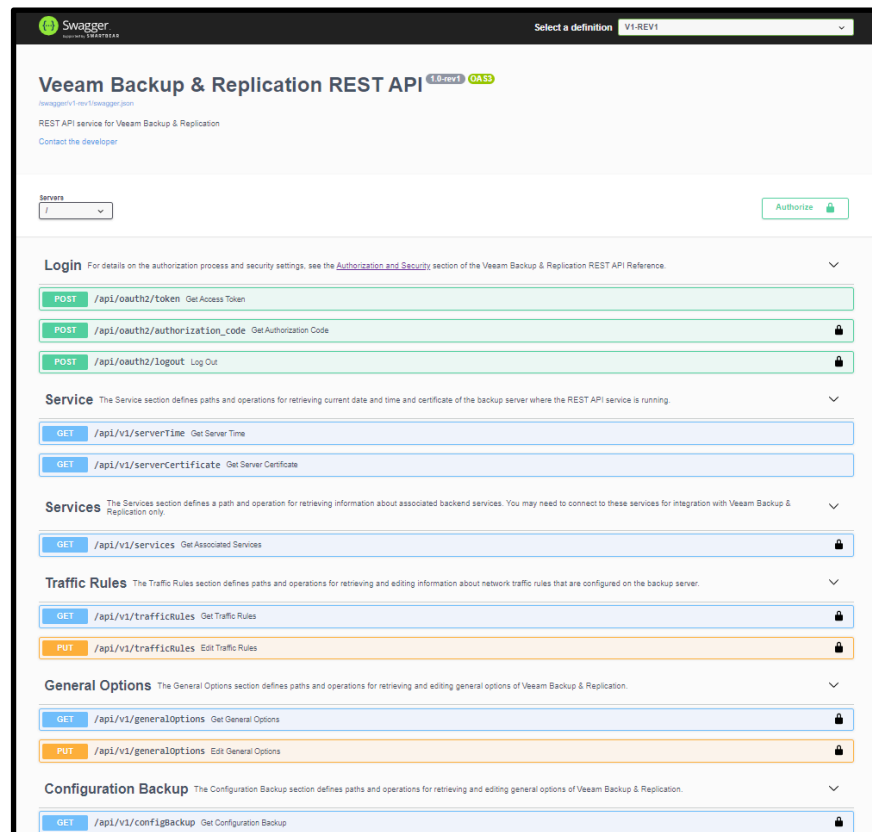


```
[default.handlers]  
error_codes = [503, 504]
```

```
[[handlers]]  
name = "/wordCounter_GET"  
type = "REST"  
latency = "100ms"  
error_codes = [500, 400]
```

SERVICE API

If you need to rich existing service answer, service API will help to understand how to do it efficiently



DEVELOP

- ▶ Product description
- ▶ Dependencies and Consumers API
- ▶ Deploy plan
- ▶ Integration Guide
- ▶ SLO\SLA\NFR
- ▶ TDR\C4 Model



PRODUCT DESCRIPTION

01. Why this service exists?

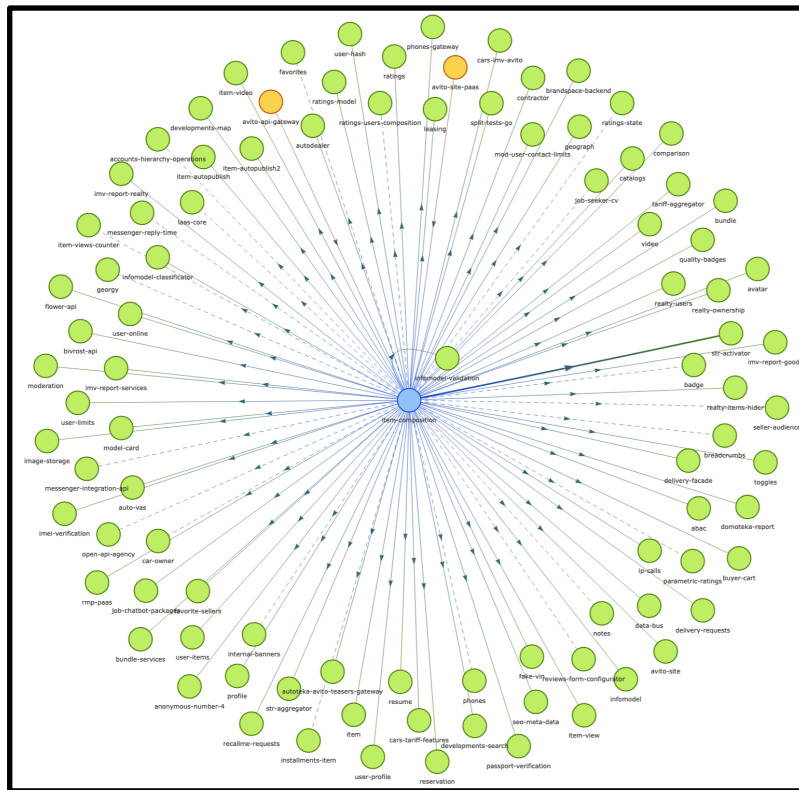
03. Who use it? What Platforms?
On what screens?

02. What problems it solves

04. What happens if derive will
down? Does it use on user crit
scenarios?

Dependencies and Consumers API

- ▶ Help with troubleshoot
- ▶ Very helpful for on-call
- ▶ Architecture optimisation (analysis)



DEPLOY PLAN AND ROLLBACK

- ▶ Rules and conditions of deploy
- ▶ Places for deploy
- ▶ Required resources
- ▶ Troubles and nuances of deploy

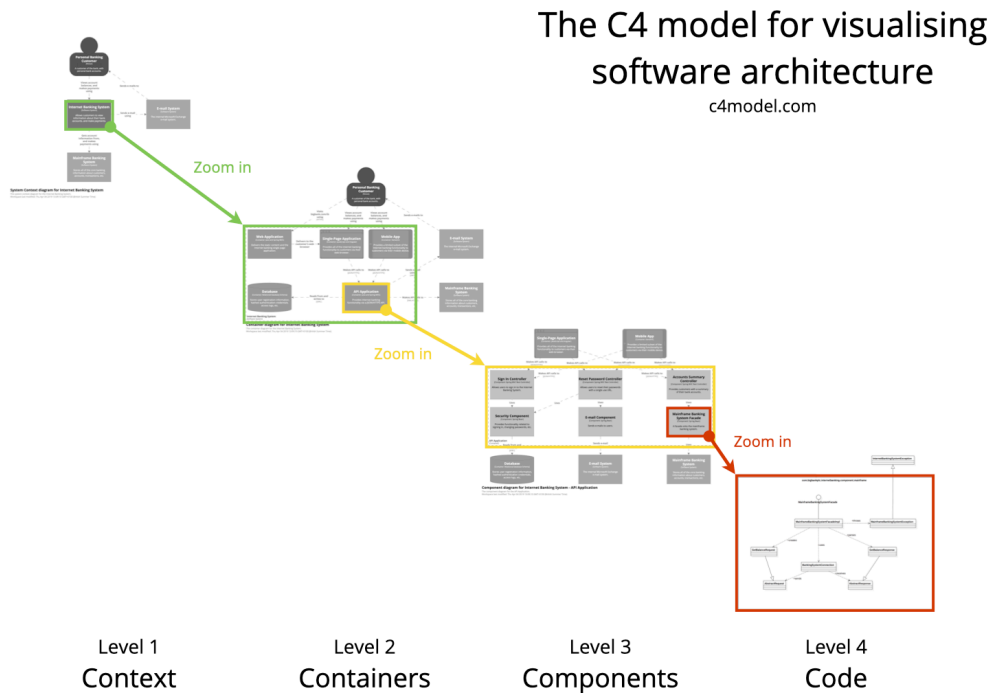


C4-MODEL

You can use any approach for visualisation of architect design

We use:

- ▶ C4 Model
- ▶ Architecture Design Record

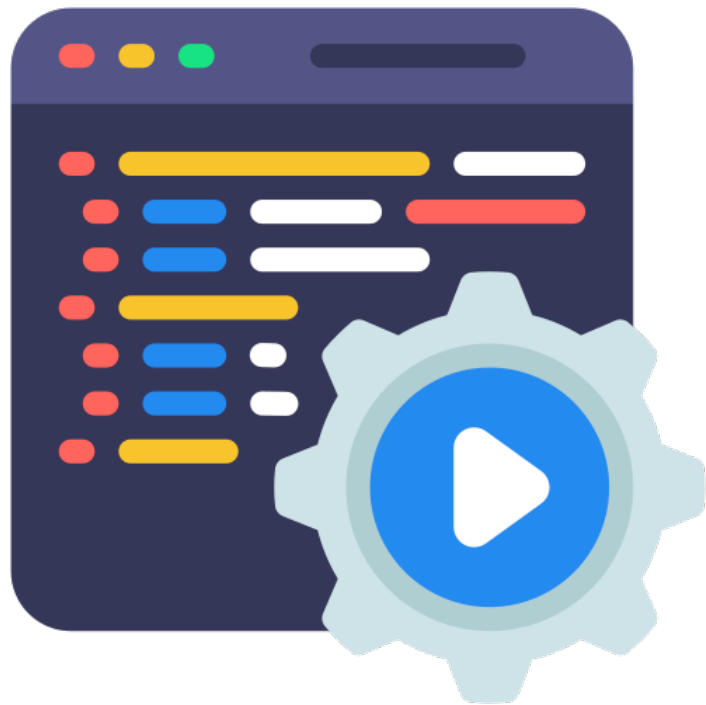


<https://c4model.com>

RUNTIME

You are on-call engineer, and you need some kind of service troubleshoot

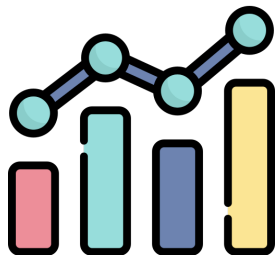
- ▶ Service observability links
- ▶ Latest releases
- ▶ Dependencies and Consumers
- ▶ Rollback plan



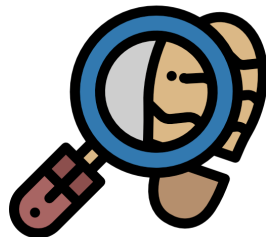
SERVICE OBSERVABILITY



Links to Logs
(Kibana,
Sentry)



Main metrics:
RPM,
Latency
Percentile



Traces (open
telemetry with
jaeger)



Links to ALERTS

Alerts.yaml in git
for every service



triggers:

- name: Panics on service

id: 5f5f0e45-9b74-42ac-91f4-09eb25a5b1d3

desc: >-

Instruction - how this problem can be solved

targets:

- aliasByNode(graphiteBucke.api.panic.total, -2, -1)

t1l_state: OK

t1l: 120

expression: "(t1 > 2) ? ERROR : OK"

dashboard: <https://link-tografana-dashboard>

pending_interval: 60

- name: 5xx in answers

desc: >-

Instruction - how this problem can be solved

targets:

- aliasByNode(movingAverage(sortByMaxima(exclude(keepLastValue(path-to-service.*.request_time.
{5*,499}.count, 1), '_info_GET'))), '1min'), 5, 7)

t1l_state: OK

t1l: 600





























expression: "t1 > 1000 ? ERROR : OK"

























dashboard: <https://link-tografana-dashboard>

pending_interval: 300









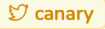
















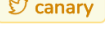




version	service	release name	strategy	time	release engineer	connected tasks
✓ ver.254	layout	main	canary	3C 13 дек. 17:49	@vc	BX-18790 ⋮
✓ ver.253	layout	main	canary	3C 12 дек. 11:08	@gs	RAR-4473 ⋮
○ ver.252	layout	main	canary	3C 2 дек. 19:17	@av	PROT-15077 ⋮
✓ ver.251	layout	main	canary	3C 30 нояб. 14:46	@vs	TM-2302 ⋮
✓ ver.250	layout	main	canary	3C 29 нояб. 14:57	@av	REAL-18680 ⋮
✓ ver.249	layout	main	canary	3C 29 нояб. 13:12	@av	PROT-15077 ⋮
✓ ver.248	layout	main	simple	3C! 23 нояб. 13:01	@ac	ANTY-2762 ⋮
✓ ver.247	layout	main	canary	4C 7 нояб. 16:30	@dv	ANTY-2762 ⋮
✓ ver.246	layout	main	canary	4C 7 нояб. 15:21	@dv	ANTY-2762 ⋮
✓ ver.245	layout	main	canary	4C 7 нояб. 13:43	@dv	SLA-1021 ⋮
✓ ver.244	layout	main	canary	4C 28 окт. 12:42	@ali	VRF-2408 ⋮







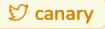
















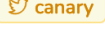




version	service	release name	strategy	time	release engineer	connected tasks
✓ ver.254	layout	main	 canary	3C 13 дек. 17:49	@vc	BX-18790   ⋮
✓ ver.253	layout	main	 canary	3C 12 дек. 11:08	@gs	RAR-4473   ⋮
✓ ver.252	layout	main	 canary	3C 2 дек. 19:17	@av	PROT-15077  ⋮
✓ ver.251	layout	main	 canary	3C 30 нояб. 14:46	@vs	TM-2302  ⋮
✓ ver.250	layout	main	 canary	3C 29 нояб. 14:57	@av	REAL-18680  ⋮
✓ ver.249	layout	main	 canary	3C 29 нояб. 13:12	@av	PROT-15077   ⋮
✓ ver.248	layout	main	 simple 	3C! 23 нояб. 13:01	@ac	ANTY-2762  ⋮
✓ ver.247	layout	main	 canary 	4C 7 нояб. 16:30	@dv	ANTY-2762  ⋮
✓ ver.246	layout	main	 canary	4C 7 нояб. 15:21	@dv	ANTY-2762  ⋮
✓ ver.245	layout	main	 canary	4C 7 нояб. 13:43	@dv	SLA-1021   ⋮
✓ ver.244	layout	main	 canary	4C 28 окт. 12:42	@ali	VRF-2408  ⋮

version	service	release name	strategy		time	release engineer	connected tasks		
✓ ver.254	layout	main	 canary	3C	13 дек. 17:49	@vc	BX-18790	+	 ⋮
✓ ver.253	layout	main	 canary	3C	12 дек. 11:08	@gs	RAR-4473	+	 ⋮
✓ ver.252	layout	main	 canary	3C	2 дек. 19:17	@av	PROT-15077		 ⋮
✓ ver.251	layout	main	 canary	3C	30 нояб. 14:46	@vs	TM-2302		 ⋮
✓ ver.250	layout	main	 canary	3C	29 нояб. 14:57	@av	REAL-18680		 ⋮
✓ ver.249	layout	main	 canary	3C	29 нояб. 13:12	@av	PROT-15077	+	 ⋮
✓ ver.248	layout	main	 simple 	3C!	23 нояб. 13:01	@ac	ANTY-2762		 ⋮
✓ ver.247	layout	main	 canary 	4C	7 нояб. 16:30	@dv	ANTY-2762		 ⋮
✓ ver.246	layout	main	 canary	4C	7 нояб. 15:21	@dv	ANTY-2762		 ⋮
✓ ver.245	layout	main	 canary	4C	7 нояб. 13:43	@dv	SLA-1021	+	 ⋮
✓ ver.244	layout	main	 canary	4C	28 окт. 12:42	@ali	VRF-2408		 ⋮



version	service	release name	strategy	time	release engineer	connected tasks
✓ ver.254	layout	main	canary	3C 13 дек. 17:49	@vc	BX-18790 ⋮
✓ ver.253	layout	main	canary	3C 12 дек. 11:08	@gs	RAR-4473 ⋮
✓ ver.252	layout	main	canary	3C 2 дек. 19:17	@av	PROT-15077 ⋮
✓ ver.251	layout	main	canary	3C 30 нояб. 14:46	@vs	TM-2302 ⋮
✓ ver.250	layout	main	canary	3C 29 нояб. 14:57	@av	REAL-18680 ⋮
✓ ver.249	layout	main	canary	3C 29 нояб. 13:12	@av	PROT-15077 ⋮
✓ ver.248	layout	main	simple	3C! 23 нояб. 13:01	@ac	ANTY-2762 ⋮
✓ ver.247	layout	main	canary	4C 7 нояб. 16:30	@dv	ANTY-2762 ⋮
✓ ver.246	layout	main	canary	4C 7 нояб. 15:21	@dv	ANTY-2762 ⋮
✓ ver.245	layout	main	canary	4C 7 нояб. 13:43	@dv	SLA-1021 ⋮
✓ ver.244	layout	main	canary	4C 28 окт. 12:42	@ali	VRF-2408 ⋮

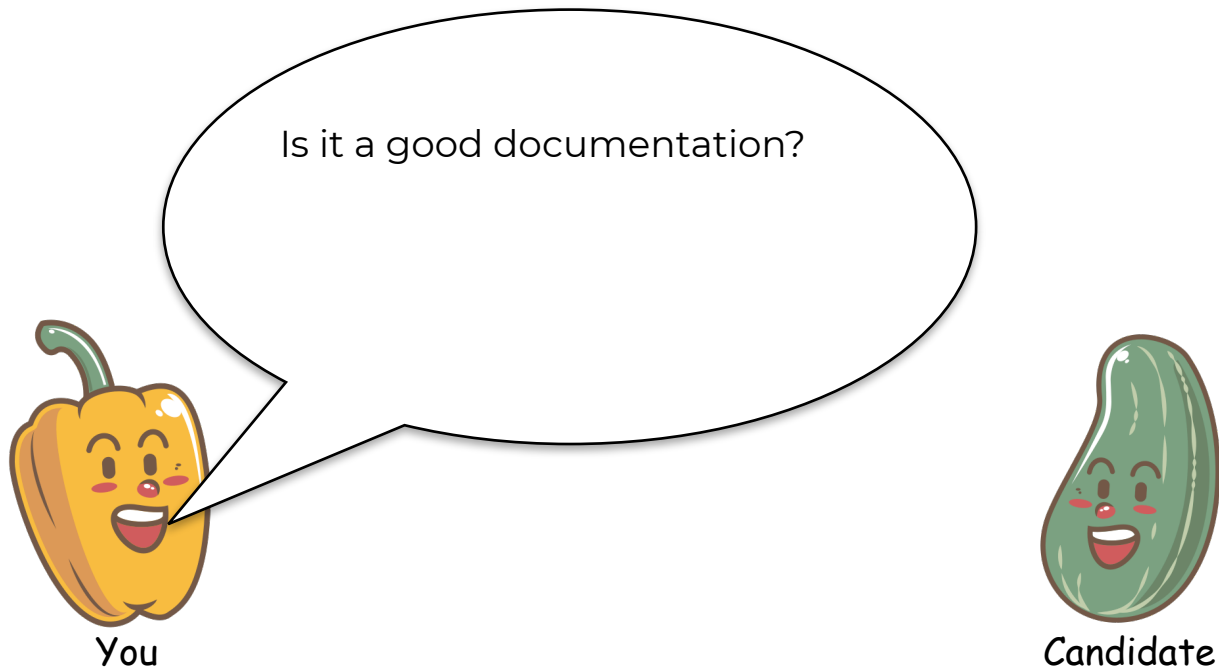
version	service	release name	strategy	time	release engineer	connected tasks
✓ ver.254	layout	main	 canary	3C 13 дек. 17:49	@vc	BX-18790   ⋮
✓ ver.253	layout	main	 canary	3C 12 дек. 11:08	@gs	RAR-4473   ⋮
✓ ver.252	layout	main	 canary	3C 2 дек. 19:17	@av	PROT-15077  ⋮
✓ ver.251	layout	main	 canary	3C 30 нояб. 14:46	@vs	TM-2302  ⋮
✓ ver.250	layout	main	 canary	3C 29 нояб. 14:57	@av	REAL-18680  ⋮
✓ ver.249	layout	main	 canary	3C 29 нояб. 13:12	@av	PROT-15077   ⋮
✓ ver.248	layout	main	 simple 	3C! 23 нояб. 13:01	@ac	ANTY-2762  ⋮
✓ ver.247	layout	main	 canary 	4C 7 нояб. 16:30	@dv	ANTY-2762  ⋮
✓ ver.246	layout	main	 canary	4C 7 нояб. 15:21	@dv	ANTY-2762  ⋮
✓ ver.245	layout	main	 canary	4C 7 нояб. 13:43	@dv	SLA-1021   ⋮
✓ ver.244	layout	main	 canary	4C 28 окт. 12:42	@ali	VRF-2408  ⋮

version	service	release name	strategy	time	release engineer	connected tasks
✓ ver.254	layout	main	 canary	3C 13 дек. 17:49	@vc	BX-18790   ⋮
✓ ver.253	layout	main	 canary	3C 12 дек. 11:08	@gs	RAR-4473   ⋮
✓ ver.252	layout	main	 canary	3C 2 дек. 19:17	@av	PROT-15077  ⋮
✓ ver.251	layout	main	 canary	3C 30 нояб. 14:46	@vs	TM-2302  ⋮
✓ ver.250	layout	main	 canary	3C 29 нояб. 14:57	@av	REAL-18680  ⋮
✓ ver.249	layout	main	 canary	3C 29 нояб. 13:12	@av	PROT-15077   ⋮
✓ ver.248	layout	main	 simple 	3C! 23 нояб. 13:01	@ac	ANTY-2762  ⋮
✓ ver.247	layout	main	 canary 	4C 7 нояб. 16:30	@dv	ANTY-2762  ⋮
✓ ver.246	layout	main	 canary	4C 7 нояб. 15:21	@dv	ANTY-2762  ⋮
✓ ver.245	layout	main	 canary	4C 7 нояб. 13:43	@dv	SLA-1021   ⋮
✓ ver.244	layout	main	 canary	4C 28 окт. 12:42	@ali	VRF-2408  ⋮



It's very important to do it all....

ON INTERVIEW



Different ways of documentation process, from service to service



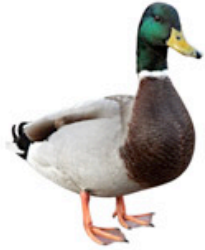
*Hooded
Merganser Duck*



*Madagascar
Teal Duck*



*Eurasian
Teal Duck*



Mallard Duck



Muscovy Duck



*Northern
Pintail Duck*



*King Eider
Duck*



*American
Black Duck*



*Red-breasted
Merganser Duck*



*Common
Merganser Duck*



Mandarin Duck



Wood Duck



Redhead Duck



Gadwall Duck



Long-tailed Duck



*Greater Scaup
Duck*



*Common
Goldeneye Duck*



Cape Teal Duck



Bufflehead Duck



*Red-billed
Teal Duck*



Harlequin Duck



Hooded
Merganser Duck



Madagascar
Teal Duck



Eurasian
Teal Duck



Mallard Duck



Muscovy Duck



Northern
Pintail Duck



Red-billed
Duck



American
Black Duck



Red-breasted
Merganser Duck

TEMPLATIZE IT



Common
Merganser Duck



Mandarin Duck



Wood Duck



Redhead Duck



Gadwall Duck



Long-tailed Duck



Greater Scaup
Duck



Common
Goldeneye Duck



Cape Teal Duck



Bufflehead Duck



Red-billed
Teal Duck



Harlequin Duck

О сервисе	
тип	business
описание	бэкенд для шапки/футера/сайдбара/поисковой строки сайта
engine	golang,1.18
юнит	AntimonoLith
ответственный	@gazdanovich
канал поддержки	~antimonoLith
теги	-- нет тегов --
тестов. покрытие	<div><div></div></div> 61.47% (mutation score: 55.8%)
NFR	.toml middleware 99.98 >= 99.90
активные тогглы	"basic_profile_enabled", "shop_in_profile_management", "stoc... история настроить

Последние релизы		<div>zeta</div> <div>hopeful-music</div> <div>deep-forest</div>		main	
версия	релизер	стратегия	время	связанные задачи	
ver.254	@v	canary	3С 17:49	BX-18790	[Front, Back] Раскатка АБ по точке входа...
ver.253	@g	canary	3С Вчера в 11:08	RAR-4473	[Backend] Исправить отображение рейт...
ver.252	@a	canary	3С 2 дек. 19:17	PROT-15077	Скрыть кнопку "запрос звонка" из Расш...
ver.251	@v	canary	3С 30 нояб. 14:46	TM-2302	[avito-site, service-layout] Добавить ссыл...
ver.250	@a	canary	3С 29 нояб. 14:57	REAL-18680	Убрать бейдж "Новое" у пункта меню "И...
ver.249	@a	canary	3С 29 нояб. 13:12	PROT-15077	Скрыть кнопку "запрос звонка" из Расш...
ver.248	@a	simple	3С 23 нояб. 13:01	ANTY-2762	Удалить hasProfessionalsNotification из с...
ver.247	@d	canary	4С 7 нояб. 16:30	ANTY-2762	Удалить hasProfessionalsNotification из с...
ver.246	@d	canary	4С 7 нояб. 15:21	ANTY-2762	Удалить hasProfessionalsNotification из с...
ver.245	@d	canary	4С 7 нояб. 13:43	SLA-1021	Добавить в nanaki mocks.coverignore

Ресурсы

34.6 RV

CPU: 30.11 спу
Память: 14.5 GB

[подробнее](#)

Oct 2022 Nov 2022 Dec 2022

Дежурный

main [@dvlukyantsev](#)

Связи

потребители

зависимости

abac

accounts-hierarchy

asd-manager

autosearch

avatar

billing-avito-gateway

слаб.

contractor

data-bus

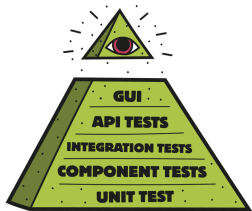
redis

grafana

MAKE UNIFY SERVICE CARD

Make it look and feel unify for all services

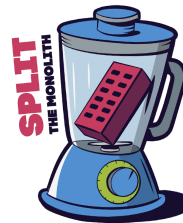
BENEFITS



Simple
navigation

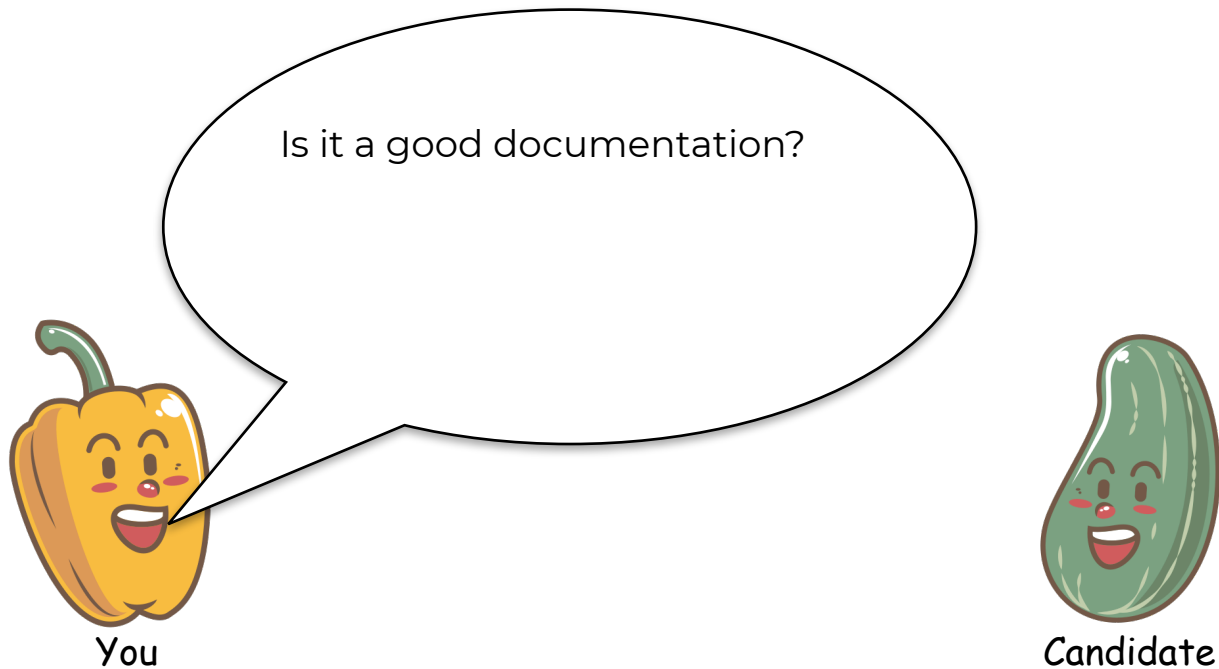


Good search
between services



Unify user experience

ON INTERVIEW





Custom
deploys is hard
to support.
Works
different from
service to
service



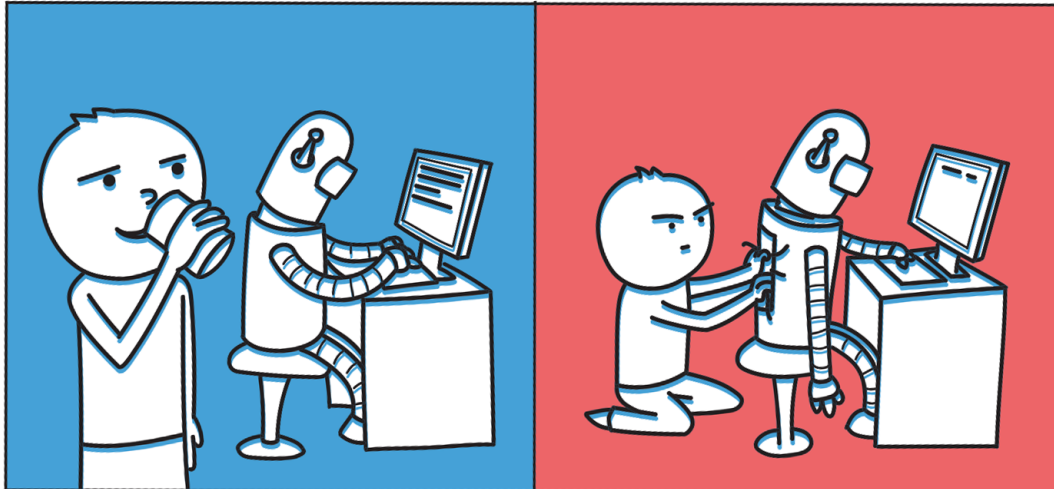
Metrics between
services are not
consistent! It's
hard to match
metrics from
one service to
another



Manually filled
release list leads
us to mistakes.



Manually
supported
service API and
links will be
eventually
inconsistent



Unification and automation for all processes!

UNIFY DEPLOY FLOW

Let everyone service be deployed only in several configurations. And only on unify CD flow with simple customisation

- ▶ Every release is simple, it's automatically register in release lists
- ▶ You can implement different release strategies (gradual, canary, etc)
- ▶ Every service will have unify rollback button



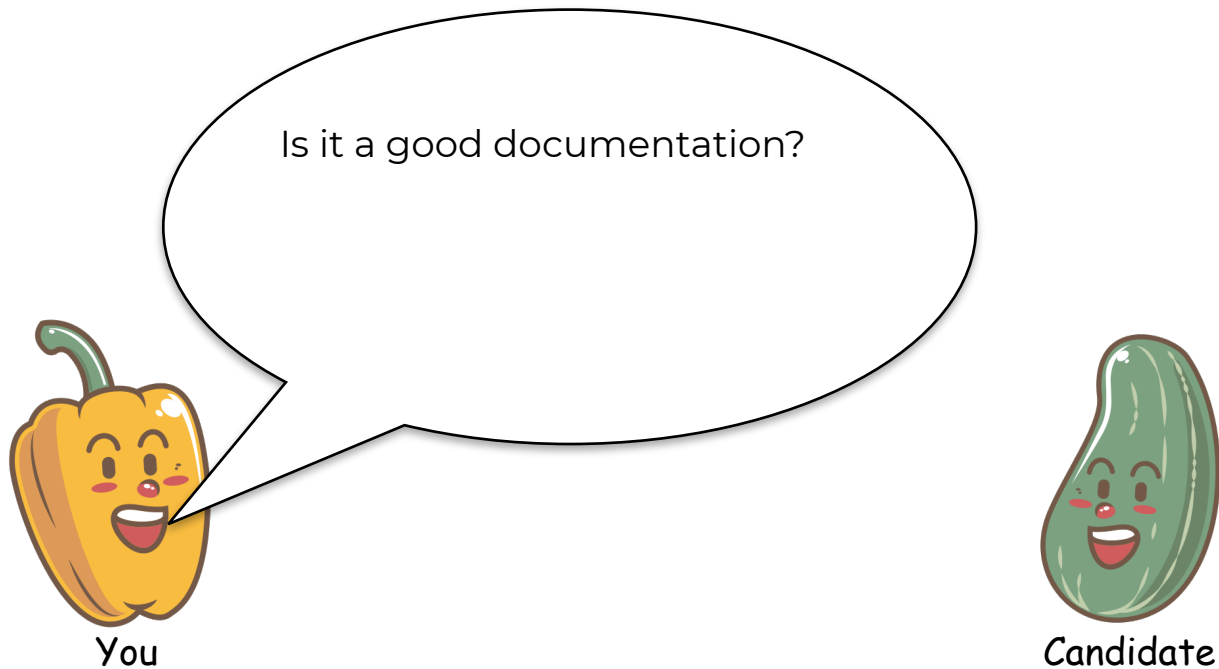
UNIFY METRICS

Create standart metrics for rpc\rest\async requests.
Implement «golden metrics» in standard buckets

- ▶ Easy to create canary strategy
- ▶ Unify alerts for similar services
- ▶ Every service will have unify rollback button



ON INTERVIEW





MEGA-important to make contract system, based on **contract-first** principles!

It helps your API-contract documentation part to be actual and consistent

CONTRACT FIRST: BENEFITS

- 01. Teams can develop new features in parallel, without lock on each other
- 02. Teams gets exactly what they are waiting for. No missing field or typo errors
- 03. Codegen-part will help you make cross platform and cross language compatibility
- 04. Scheme reuse helps decrease copy-paste and increase efficiency.

CONTRACT FIRST: LIMITATIONS

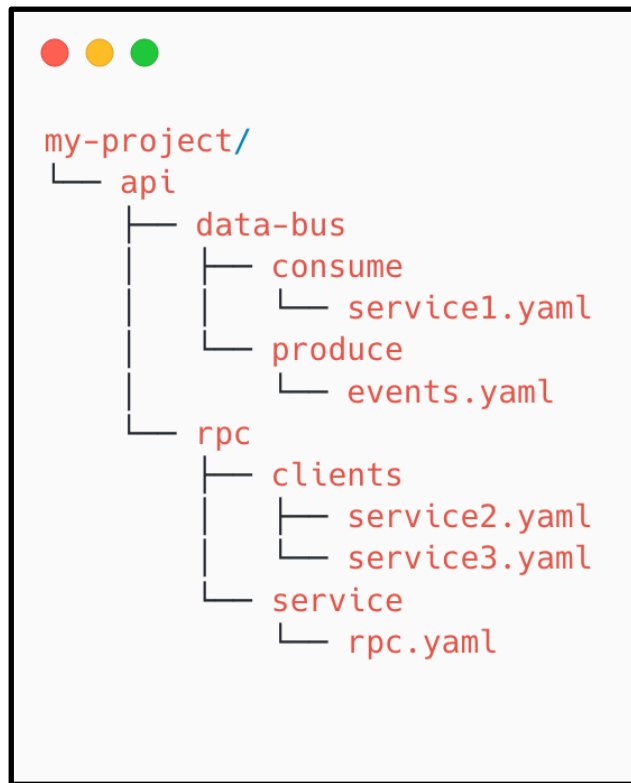
- 01. You should create standards for your internal and external API
- 02. It's expensive at the first time
- 03. You will had to create contract update and validate system



Your contract documentation will be always actual and full!

CONTRACT FIRST: CREATING CONTRACTS

- ▶ Describe service new method in [*rpc.yml*](#)
- ▶ Describe input, output schema for new method
- ▶ Run [*codegen*](#) command, for creating empty handler and to for new method in service



CONTRACT FIRST: FIRST DEPLOY

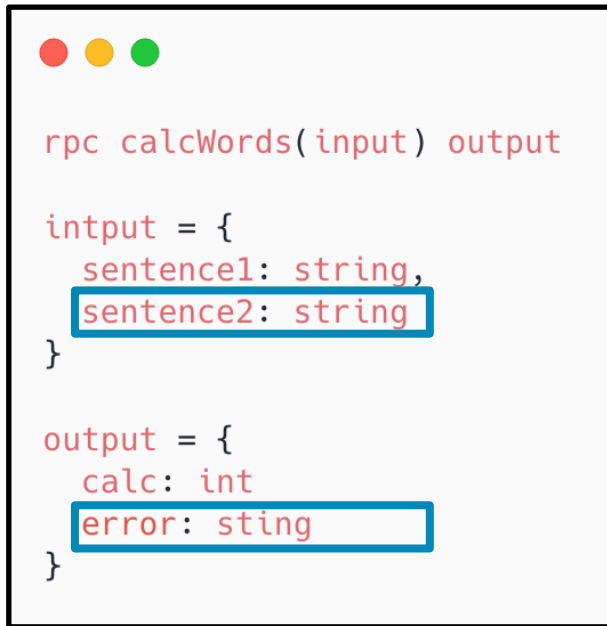
- ▶ New method will be register in schema-storage
- ▶ Changes in old method will be validate for backward compatibility with consumers, using that method
- ▶ After success validation modify method will be updated in schema-storage

Validation

Will check every deploy

- ▶ All method consumers input against service schema
- ▶ All consumers output against service schema

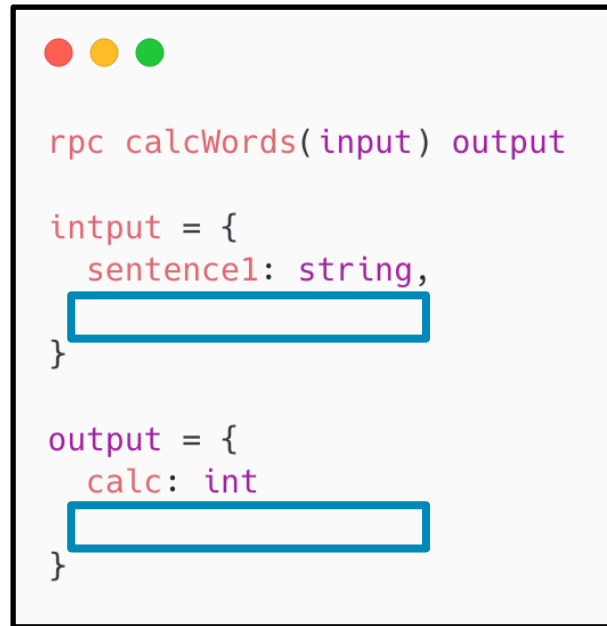
CONTRACT FIRST: FIRST DEPLOY



SERVICE



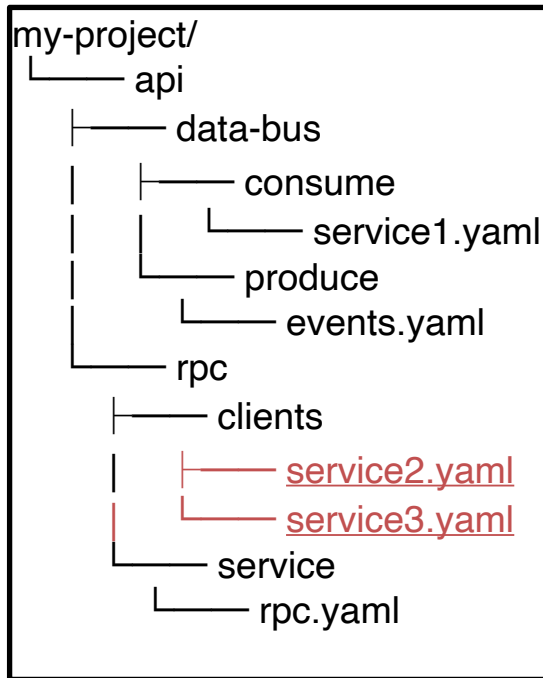
Check backward
Compatibility



CONSUMER

CONTRACT FIRST: CONSUMER ADD DEP

- ▶ Describe **exact** method, which needs.
- ▶ Describe exact fields for method output
- ▶ Run [*codegen*](#) command, for creating empty handler and to for new method in service



```
rpc calcWords(input) output

input = {
  sentence1: string,
}

output = {
  calc: int
}
```

CONTRACT FIRST: CONSUMER DEPLOY

- ▶ Validation newly added service
- ▶ Describe exact fields for method output
- ▶ Run [codegen](#) command, for creating empty handler and to for new method in service

```
rpc calcWords(input) output

input = {
  sentence1: string,
  sentence2: string
}

output = {
  calc: int
  error: sting
}
```

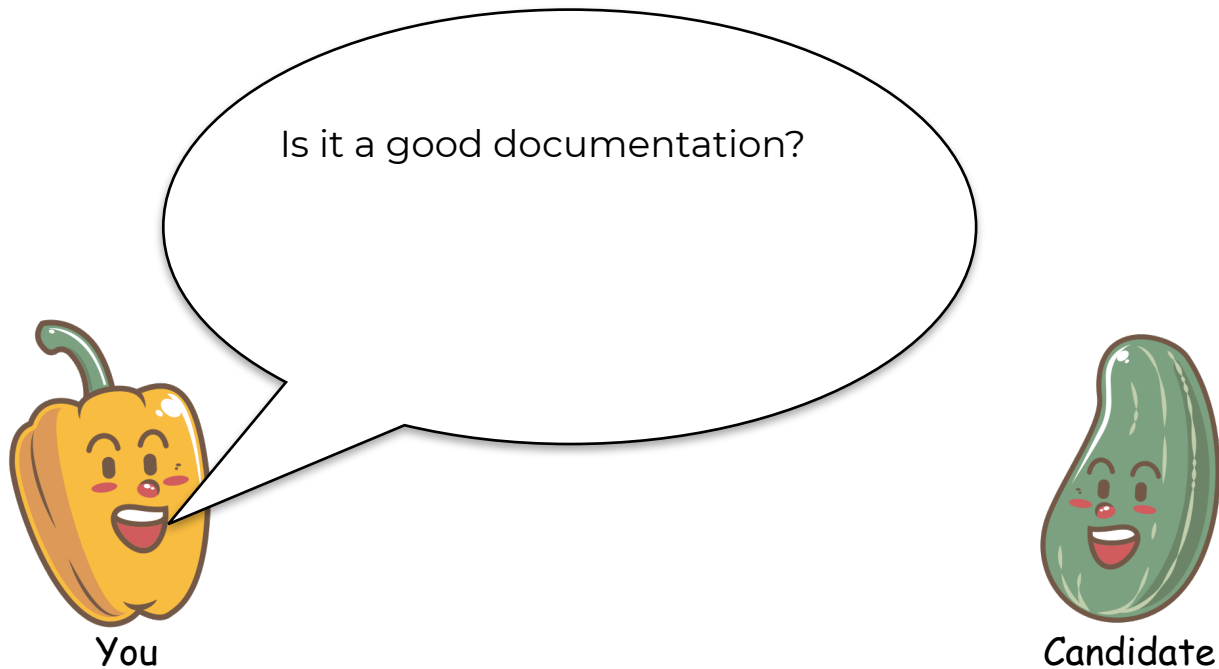
Consumer schema
Should be subset
Of service

```
rpc calcWords(input) output

input = {
  sentence1: string,
}

output = {
  calc: int
}
```

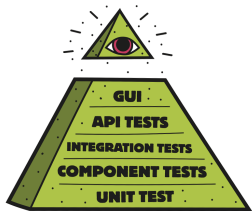
ON INTERVIEW



A man with a mustache and sunglasses, wearing a white tank top, is shown from the chest up. His right arm is raised, and water droplets are visible on his skin. A white speech bubble with a black outline is positioned to the left of his head, containing the text 'MkDocs'. The background is dark and out of focus, suggesting an outdoor setting at night.

MkDocs

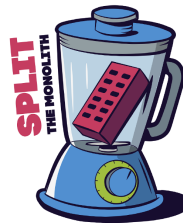
MkDocs



Is part of code,
described as
usual markdown



Is deployed with
service on
[codegen](#) phase.



Can be unite in one
documentation for all
service

ON INTERVIEW



01. Now we have universal description for microservices

02. Now we have links to all significant runtime metrics

03. We have self-supported actual and fresh API documentation

04. We have in-code generated documentation



avito.tech

Yerevan — 2022

Pavel Lakosnikov

SLA TeamLead in Avito



twitter.com/shaman_s_bubnom



t.me/shaman_s_bubnom

Links:

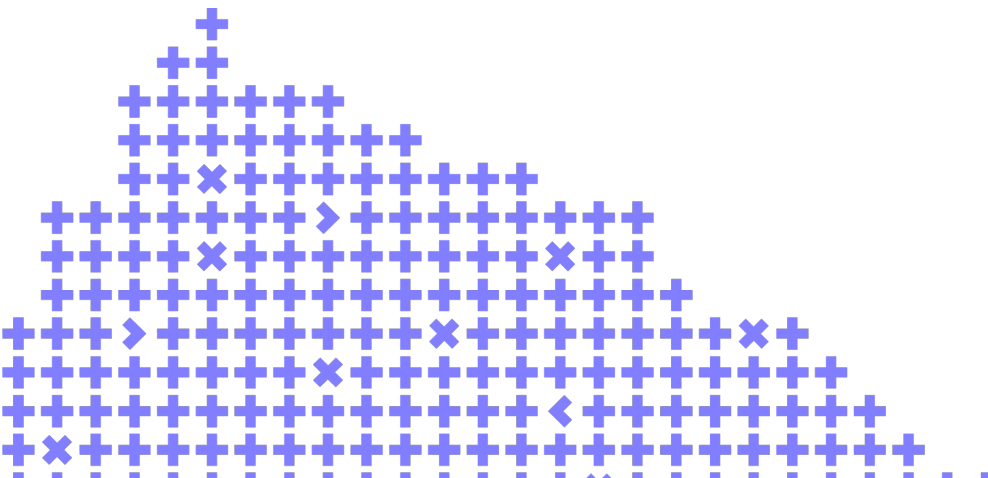
cortex.io/post/how-to-strategically-approach-documenting-microservices

c4model.com

swagger.io/resources/articles/adopting-an-api-first-approach/

Leave your feedback!

You can rate the talk and give
a feedback on what you've
liked or what could be
improved



Co-organizer

Yandex